DUPLICATE

Forecasting Research

Forecasting Research Division
Technical Report No. 147

Macrotasking the Atmospheric Prediction Component of the Unified Model

by

A. Dickinson and D.J. Salmond

December 1994

Meteorological Office
London Road
Bracknell
Berkshire
RG12 2SZ
United Kingdom

ORGS UKMO F
National Meteorological Library
FitzRoy Road, Exeter, Devon. EX1 3PB

Forecasting Research Technical Note No 147

Macrotasking the Atmospheric Prediction Component of the Unified Model

A. Dickinson and D. J. Salmond*

December 1994

* Cray Research, Bracknell, UK

This document has not been published. Permission to quote from it must be obtained from the Director of Forecasting Research.

© Crown Copyright 1995

1. Introduction

The Met Office has an ongoing requirement to reduce the run time of its NWP models so that improvements in formulation can be accommodated within the time window available to run an operational forecast. Almost without exception, formulation enhancements require an increase in computer resources. This can be seen from the model development plans for 1995, which include a 50% increase in the number of levels used in global and LAM configurations of the Unified Model (UM) and the introduction of a more accurate (but more expensive) radiation scheme. The Met Office is also beginning to prepare for a major enhancement to its current supercomputer, a 16-processor CRAY C90, which will be delivered late in 1996. The architecture of the new machine is not certain at this stage, but it is very likely that it will be a distributed memory massively parallel processing (MPP) system. The statement of requirement for this computer will be based on demonstrating a significant increase in throughput relative to the CRAY C90. It is therefore important to optimise the efficiency of the UM on the C90 (so that a true measure of speed-up is obtained) and to begin the process of restructuring the code so as to make it more suitable for running on an MPP system.

This paper describes improvements to the parallel structure of the atmospheric prediction component of the Unified Model which make the code more efficient when run on a CRAY C90. The changes affect both the dynamics and the physics of the model and have been incorporated into the model build at Version 3.4. Up until now, parallelisation of the UM on Cray computers has been achieved by using the pre-compiler steps *fpp* and *fmp* to implement *Autotasking* of inner loops. This technique generates a separate parallel region for each DO loop. An alternative approach which uses larger parallel regions containing many DO loops is found to be more efficient on the C90. The term *macrotasking* will be used in the remainder of this paper to describe this approach.

The impact of the *macrotasking* approach is illustrated using a set of integrations based on standard climate, global forecast and mesoscale experiments at Version 3.2 of the UM. Interpretation of these results is aided by the use of a simple advection testbed program. This program integrates the atmospheric model's advection routines from artificial initial data using a uniform grid length over a limited area domain. Full details of the formulation of the split-explicit finite-difference integration scheme currently used in the atmospheric model are given by Cullen and Davies (1991).

The array dimensions used in the various model configurations determine the efficiency of the UM under different parallelisation scenarios. Table 1 lists the array sizes in use at the present time.

Configuration	East-West	North-South	Levels Now	Levels Future
Climate	96	73	19	31?
Global Forecast	288	217	19	31
LAM	229	137	19	31
Mesoscale	92	92	31	31

Table 1: Array dimensions used by various Unified Model configurations. An increase in the number of levels used by the large-scale operational configurations is expected during 1995.

2. Parallelisation strategies

2.1 Autotasking of inner loops

The atmospheric model's data structure consists of a set of horizontal fields organised by levels. Each subroutine has an outer loop going over the model levels which contains a number of inner loops each spanning a horizontal field. *Autotasking* of the innermost loops (sometimes referred to as 'stripmining') was chosen for the first implementation of the atmospheric component of the UM on an 8 CPU, 32 Mword CRAY Y-MP, since it offered good levels of parallelisation, particularly at global forecast resolution, without the need to consider any horizontal or vertical dependencies. This parallelism was implemented automatically by the compiling system. Its use implied no memory overheads and also gave excellent load balancing. (Dickinson, 1990)

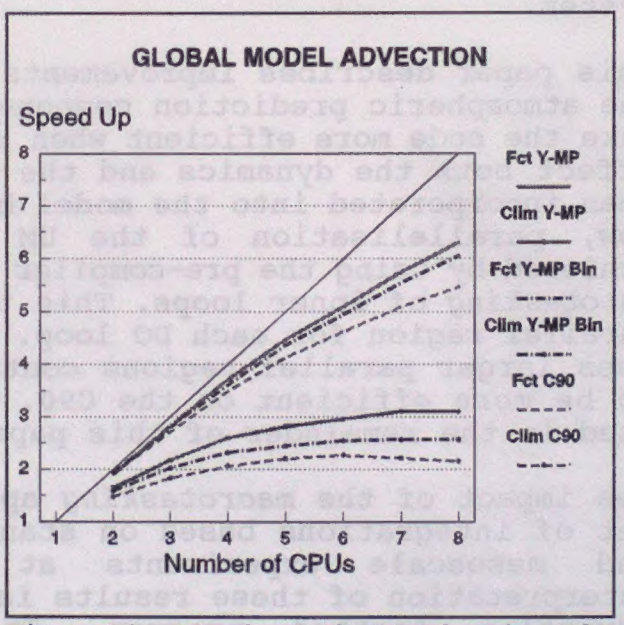

Figure 1: Speed up observed using the advection testbed run using climate (Clim) and global forecast (Fct) dimensions on (a) a CRAY Y-MP, (b) a CRAY C90, and (c) a CRAY C90 using a CRAY Y-MP binary.

Table 2 gives details of the Y-MP and C90 computers installed in the Met Office. Each processor on the CRAY C90 is potentially 3 times faster than on the Y-MP (that is 1.5 times for the ratio of the clock speeds and a further 2 times as there are twice as many vector pipes). Significant reductions in run time have been observed since the replacement of the Y-MP by a C90 in May 1994. It is therefore surprising to discover that the atmospheric model is in fact less parallel on the C90 than on the

Y-MP. This is illustrated in Figure 1. As can be seen, the speed of the advection routines scales relatively poorly with the increase in the number of CPU's on the C90. In the case of those configurations with smaller horizontal dimensions, such as the climate configuration, there is a characteristic peak and then reduction in speed up as the number of processors is increased. The reasons for this are manyfold. Firstly, although the peak vector performance increased by a factor of 3 between the Y-MP and the C90, the scalar performance only increased by a factor of 1.5. Since each parallel region is preceded by a section of scalar code which initiates and controls the parallelisation, the cost of this "start up" is relatively greater on the C90. Secondly, the length of the vector registers have been increased from 64 elements on the Y-MP to 128 elements on the C90. The reduction in vector performance as the number of grid points per CPU decreases will therefore be more marked on the C90 than on the Y-MP.

A third effect is due to the difference in memory latency between the C90 and the Y-MP. The memory latency on the Y-MP is a smaller number of clock periods than on the C90. This further reduces the scalar speed up between the C90 and the Y-MP. Figure 1 illustrates how these three components affect the advection at global forecast and climate resolutions in the original code. The best speed-up's are obtained on the Y-MP. The effect of the different memory latencies is shown by comparing the Y-MP speed-up curve with the curve of the Y-MP binary run on the C90. This difference is smaller at global forecast resolution as the tasks are large compared with the scalar overhead. The effect of the 128 vector length (as compared with 64 vector length) and the doubling of the number of pipes can be examined by comparing the runs using the Y-MP binary on the C90 with the native C90 code. This combined effect gives a relatively smaller change at climate resolution, where the shorter loop length means that good levels of parallelism cannot be obtained even with a vector length of 64.

	Number of CPU's	Memory (Mwords)	Clock speed (nsecs)	Memory latency (clocks)	Vector length	Number of sets of vector pipes	Date
Y-MP	8	32 then 64	6	17	64	1	Dec 1990
C90	16	256	4	23	128	2	May 1994

Table 2: Cray computers installed at the Met office since 1990.

2.2 Macrotasking parallelisation strategy

The macrotasking modifications seek to reduce the number of parallel regions thereby reducing the overall cost of parallel

region start up. They also seek where possible to maintain the vector length at the size of the full horizontal field. A different strategy has been employed in the dynamics and physics sections because of the intrinsically different dependencies they exhibit. In order to minimise the impact on other users, an approach of least changes to the existing code was taken.

The following code sections were revised in the course of this exercise:

Dynamics: *Adjustment, Advection, Diffusion*
Physics: *Long Wave Radiation, Short Wave Radiation, Convection*

Other physics routines, for example the large-scale cloud and gravity wave drag schemes, can also be *macrotasked* in the same way. These routines require additional restructuring and so conversion has been delayed until a later stage.

In the dynamics, the code structure dictates that the only viable alternative to parallelising inner loops is to parallelise over levels. This approach was chosen, even though for the cases with 19 levels the degree of parallelism is fairly limited for large numbers of CPU's. The terms with vertical dependencies, such as the calculation of the vertically integrated divergence and the hydrostatic equation, continue to be calculated using the inner loop method. The remaining terms, where there are no vertical dependencies, are macrotasked over levels using the CFPP$ CNCALL directive to *fpp*, which forces the outer loop over levels to be parallelised. In most cases the scoping of variables to be either shared or local to a task is done automatically by *fpp*. In a few places, the automatic method cannot cope with the complexity of the DO loops and a CMIC$ DO ALL directive is used with all variables explicitly scoped by hand.

In the advection subroutines, changes were made to remove artificial dependencies between levels resulting from the memory saving programming strategy adopted for the 32 Mword CRAY Y-MP system when the UM was first coded. The original code had one loop over levels for the two half timesteps. This loop is now split into two independent loops with no vertical dependencies. This results in the introduction of some new 3-D arrays which significantly increase the memory used in these routines. However, the total memory used by the model increases only in the case of the global forecast resolution, where the memory high-water mark is increased by 5 Mwords. The memory high-water mark in mesoscale and climate configurations occurs in the physics routines, because a smaller number of segments is used in the radiation and convection routines in these configurations in order to provide a reasonable vector length. The splitting of the advection into two independent loops is a necessary preparatory step towards the coding of an efficient Message Passing version of the model for a distributed memory MPP system. The restructuring of these routines will allow the exchange of edge data between adjacent processors to be done for all levels at the same time, thus significantly reducing the impact of message passing latency.

In the physics, the radiation and convection subroutines employ a scheme to reduce the memory used by intermediate work arrays, called 'segmentation', where the calculations are split into a pre-specified number of horizontal segments with no dependencies between them. The macrotasking layer was therefore applied at the outer loop over the segments using the CFPP$ CNCALL directive. The level of parallelism and the scope for load balance can be controlled by the number of segments chosen, but for large numbers of segments this is at the expense of the vector length. The model detects how many CPU's are specified for the run from the environment variable, NCPUS, and calculates the number of parallel segments used to be (NCPUS)*(the number of radiation or convection segments specified via the User Interface).

3. Impact of changes

The results of the changes from *Autotasking* to *macrotasking* were investigated by running a number of tests at mesoscale, climate and global forecast resolutions on a range of numbers of CPU's. The tests were carried out on both the full model and the simple advection testbed using a dedicated CRAY C90 system sited at Cray Research's headquarters in Minneapolis, U.S.A.. A summary of the overall improvement for the full model for each configuration is shown in Table 3. Currently, 3 CPU's are used for operational runs of the model at all configurations. Climate runs often use only 1 or 2 CPU's. It is expected that 5 CPU's will be used for operational work, once all versions of the model move to using 31 levels.

The largest impact of the changes occurs in those cases where the original *Autotasking* implementation was performing least well i.e. in the mesoscale and climate configurations. In all cases the impact increases with the number of CPU's used. The *macrotasking* modifications to the dynamics show the largest improvement in the mesoscale configuration, where the parallelisation is over 31 levels, rather that the 19 levels used in the other two configurations tested. This is a direct consequence of improved load balancing and is clearly illustrated by the advection testbed results shown in Figures 3a-c. The highest efficiency occurs when the number of CPUs approaches an exact divisor of the number of levels - it cannot be an exact divisor because the number of model levels is a prime number! Figure 3c indicates that further improvements in the parallel efficiency of the global forecast and climate configurations will occur when these models move to using 31 levels.

The overall reduction in elapsed time due to the *macrotasking* modifications may be broken down by individual model sections. This is shown in Figure 2 for the mesoscale configuration using 5 CPU's. All the revised sections show a significant improvement, the least impact being in (i) the adjustment routines, where the vertical dependencies reduce the scope for macrotasking over levels and (ii) the convection scheme, where the pre-compression to active points only, gives some load imbalance between the segments. Figures 4a-c show curves of speed-up plotted against

number of CPU's for the original and macrotasked versions for each of the three configurations. More detailed timing information for the three configurations is given in Tables 4a-f at the end of the paper.

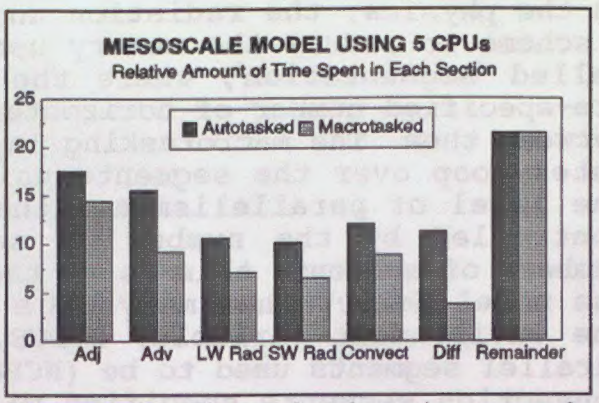

Figure 2: Elapsed time in seconds taken by revised sections.

The least impact from the changes is seen for the global forecast configuration. This is because the *Autotasking* scheme was already performing well for this case, as the amount of work for each parallel region was much larger compared with the serial overhead than in the other configurations. The short wave radiation does show a much improved speed up for the global forecast configuration. This is because the memory conflict due to several processors simultaneously accessing the same element in the look-up tables is much reduced by the introduction of the larger tasks containing a range of operations. However, the version of the long wave radiation scheme used in the global forecast model (version 2.1b using the ECMWF like transmissivities) has no look-up tables and shows only a small improvement from the use of larger tasks.

Runs of the mesoscale forecast were analysed with *Atexpert*, a CRAY tool for measuring the *Autotasked* code efficiency of a program. For a 1 hour forecast the total number of parallel regions executed was reduced from 612722 to 76146, thus reducing the cost of the parallel region start-up by a factor of 8.

	2 CPU	3 CPU	5 CPU	8 CPU	Memory Increase Mwords
Climate	5(13%)	6(15%)	7(25%)	6(34%)	0
Global Fc	18(4%)	8(3%)	11(5%)	8(5%)	5
Mesoscale + long physics timestep	84(15%)	91(18%)	110(28%)	133(38%)	0

Table 3: Reduction in elapsed time for a one day forecast resulting from the macrotasking changes. Values are in seconds followed by the percentage overall reduction. The change in memory required to run a configuration is also shown.

4. Programming bugs

A number of long standing programming bugs came to light as a result of this exercise. They are noted in passing since they

effect both the reproducibility of the code and the formulation.

The segmentation of the physics contained two minor bugs, which could give rise to different results if the number of segments used by the convection or short wave radiation schemes were changed. More seriously, it was discovered that the wrong level was being used for **etadot**, the vertical velocity relative to the model's horizontal surfaces, when calculating the vertical component of the coriolis term. The effect of correcting this error on forecast skill is only marginal and difficult to discern, but the remedy cannot be introduced into climate integrations until the Vn3.5 build, when it is planned to temporarily relax the reproducibility constraint between model builds.

5. Conclusions

The "*Autotasking* of inner loops" parallelisation strategy used in the atmospheric prediction component of the UM has been reassessed and *macrotasking* modifications introduced. These have led to a reduction in model run time in parallel mode on the CRAY C90 allowing more efficient use of larger numbers of processors. The impact is greatest for those model configurations with small horizontal dimensions such as used in climate integrations and mesoscale forecasting. The modifications maintain bit reproducibility with the previous version of the code.

The *macrotasking* version of the code does not include any inherent load balancing, which is a natural feature of the "*Autotasking* of inner loops" approach. The degree of parallelism in the dynamics is limited to the number of model levels. This can give poor load balancing when using larger numbers of CPU's, particularly with 19-level versions of the model. Improved levels of parallel efficiency will be seen once all model configurations switch to using 31-levels.

A major revision to the structure of the advection code was found to be necessary in order to remove artificial vertical dependencies resulting from the previous memory-saving coding strategy. This has led to an increase of 5 Mwords in the memory required to run global forecast configurations. Other configurations are unaffected because their memory high-water mark occurs in the physics. The restructuring of the advection routines has resulted in code more suitable for efficient running on a distributed memory MPP computer.

Future work will include the segmentation of the remaining physics routines and load balancing of the physics. This work will be carried out in conjunction with the University of Manchester.

6. Acknowledgments

The authors wish to thank the following: Rupert Ford of

Manchester University for providing help with the parallelisation of the physics routines; Mark Mawson for reviewing the dynamics changes and identifying the bug in the advection code; and Cray Research for providing access to a dedicated CRAY C90, without which it would not have been possible to measure the performance gains.

7. References

Dickinson A., 1990, Autotasking forecast and climate models on the CRAY Y-MP system. *Cray Channels*, **Vol.12, No.3.**

Cullen M.J.P., Davies T., 1991, A conservative split-explicit integration scheme with fourth order horizontal advection. *Q.J.R. Meteorol. Soc.*, **117**, 993-1002.

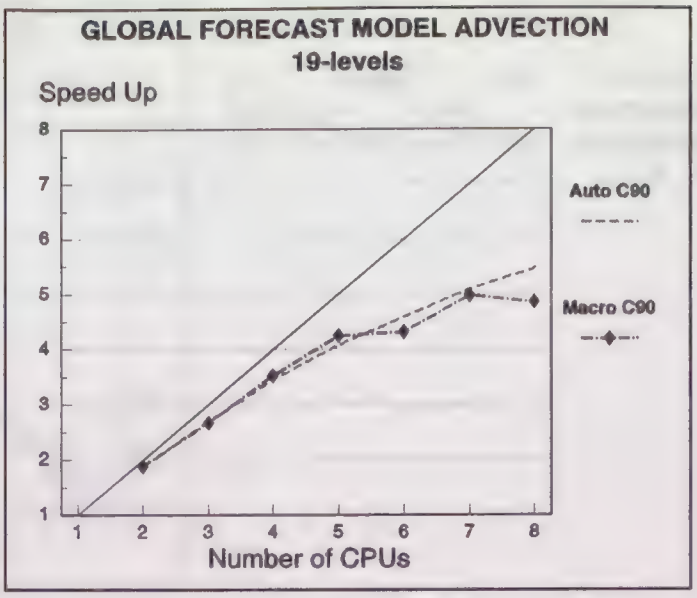

Figure 3a: Speed up curves obtained using global forecast model dimensions in advection testbed.

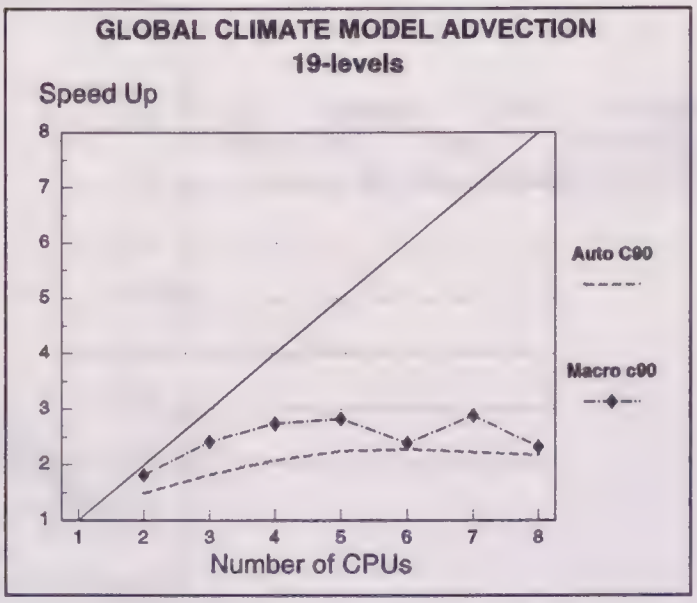

Figure 3b: Speed up curves obtained using climate model dimensions in advection testbed.

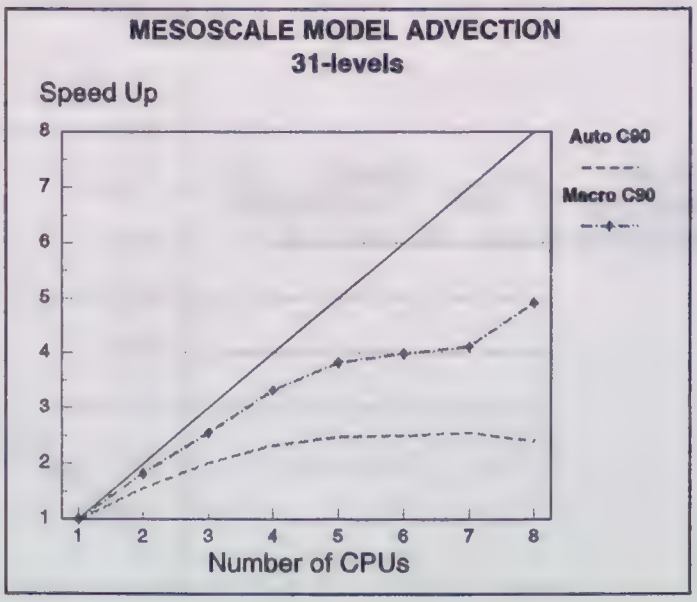

Figure 3c: Speed up curves obtained using mesoscale model dimensions in advection testbed.

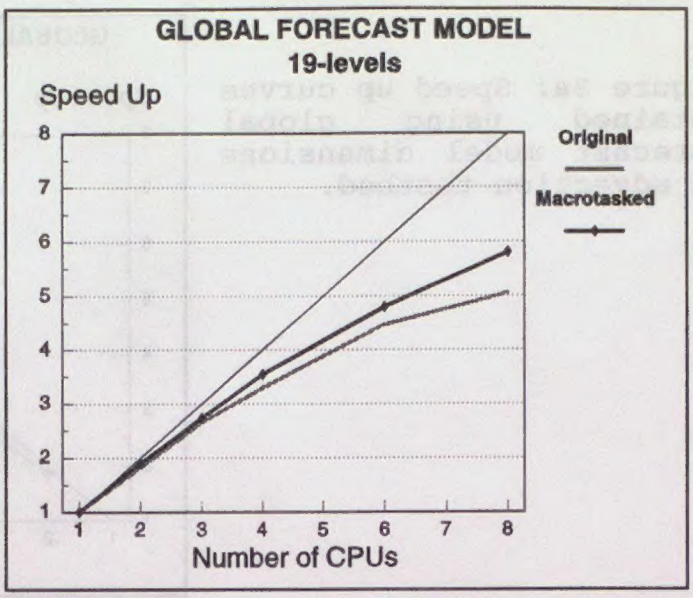

Figure 4a: Speed up observed for global forecast configuration at Vn 3.2.

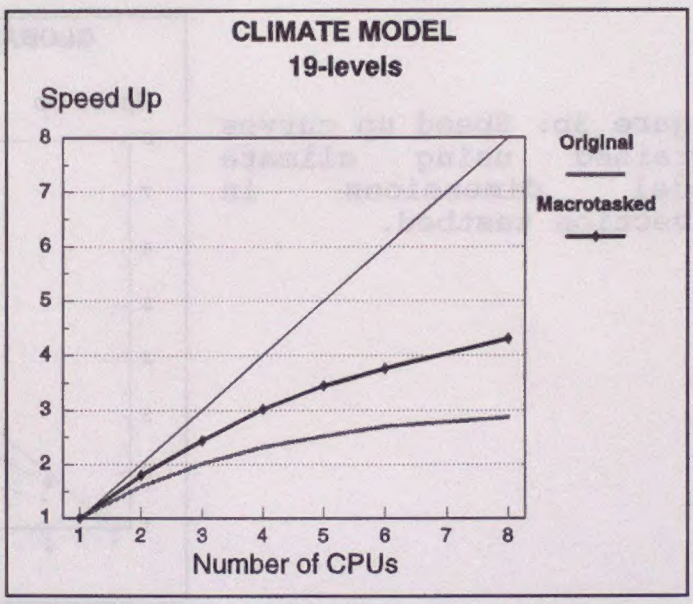

Figure 4b: Speed up observed for climate configuration at Vn 3.2.

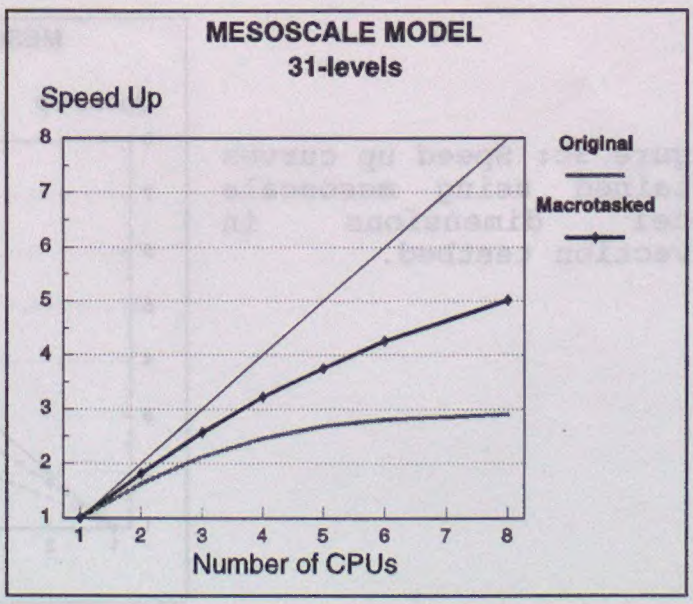

Figure 4c: Speed up observed for mesoscale configuration at Vn 3.2

Table 4a: UNIFIED MODEL (Original Version)
Dedicated runs of 3 hour Global Forecast on C916 (Sept 94) Grid = 288 x 217 x 19, Timestep = 10 minutes, Resolution = 100 km
Elapsed Times in Seconds (speed-up)

	1 CPU	2 CPUs	3 CPUs	4 CPUs	5 CPUs	6 CPUs	8 CPUs	16 CPUs
ADJ	26.54	14.23 (1.86)	9.30 (2.85)	7.32 (3.62)	5.85 (4.53)	4.94 (5.37)	4.04 (6.56)	2.63 (10.09)
ADV	16.89	8.99 (1.87)	5.99 (2.81)	4.75 (3.55)	3.84 (4.39)	3.28 (5.14)	2.73 (6.18)	1.85 (9.12)
LWRAD	12.67	6.68 (1.89)	4.41 (2.87)	3.40 (3.72)	2.76 (4.59)	2.31 (5.48)	1.84 (6.88)	1.09 (11.62)
SWRAD	8.69	4.78 (1.81)	3.45 (2.51)	2.87 (3.02)	2.56 (3.39)	2.30 (3.77)	2.11 (4.11)	2.01 (4.32)
CONVECT	10.98	6.09 (1.80)	4.13 (2.65)	3.40 (3.22)	2.81 (3.90)	2.46 (4.46)	2.11 (5.20)	1.53 (7.17)
DIFF	5.63	2.95 (1.90)	2.05 (2.74)	1.60 (3.51)	1.36 (4.13)	1.17 (4.81)	0.96 (5.86)	0.71 (7.92)
TOTAL	110.69	59.86 (1.84)	40.81 (2.71)	32.73 (3.38)	27.22 (4.06)	23.61 (4.68)	19.89 (5.56)	14.32 (7.72)
MFLOPS	472	872	1278	1592	1913	2204	2614	3622
MEMORY (MWORDS)	22.1	22.1	22.2	22.2	22.2	22.2	22.3	22.5
TOTAL CPU	109.70	116.66	119.04	125.05	128.25	133.72	148.68	216.97

Table 4b: UNIFIED MODEL (New Version with Multitasking Mods)
Dedicated runs of 3 hour Global Forecast on C916 (Sept 94) Grid = 288 x 217 x 19, Timestep = 10 minutes, Resolution = 100 km
Elapsed Times in Seconds (speed-up)

	1 CPU	2 CPUs	3 CPUs	4 CPUs	5 CPUs	6 CPUs	8 CPUs	16 CPUs
ADJ	26.63	13.91 (1.91)	9.54 (2.79)	7.24 (3.67)	5.93 (4.49)	5.44 (4.89)	4.47 (5.95)	3.23 (8.24)
ADV	15.90	8.12 (1.95)	5.74 (2.77)	4.32 (3.68)	3.77 (4.21)	3.43 (4.63)	2.98 (5.33)	4.52 (3.51)
LWRAD	12.66	6.36 (1.99)	4.26 (2.97)	3.21 (3.94)	2.58 (4.90)	2.19 (5.78)	1.63 (7.76)	0.91 (13.91)
SWRAD	8.58	4.38 (1.95)	2.96 (2.89)	2.27 (3.77)	1.85 (4.63)	1.57 (5.46)	1.21 (7.09)	0.71 (12.08)
CONVECT	10.99	5.83 (1.88)	4.02 (2.73)	3.13 (3.51)	2.55 (4.30)	2.17 (5.06)	1.70 (6.46)	1.01 (10.88)
DIFF	5.62	2.93 (1.91)	2.02 (2.78)	1.49 (3.77)	1.26 (4.46)	1.10 (5.10)	0.88 (6.38)	0.62 (9.06)
TOTAL	109.65	57.68 (1.90)	40.04 (2.73)	31.10 (3.52)	25.87 (4.23)	23.04 (4.75)	19.02 (5.76)	15.59 (7.03)
MFLOPS	480	912	1312	1688	2028	2276	2754	3354
MEMORY (MWORDS)	30.0	30.6	31.3	32.0	32.6	32.9	34.1	38.3
TOTAL CPU	108.76	111.56	113.29	115.10	116.63	119.48	123.86	145.10

Table 4c: UNIFIED MODEL (Original Version)
Dedicated runs of 1 day climate on C916 (Sept 94) Grid = 96 x 73 x 19, Timestep = 30 minutes, Resolution = 300 km
Elapsed Times in Seconds (speed-up)

	1 CPU	2 CPUs	3 CPUs	4 CPUs	5 CPUs	6 CPUs	8 CPUs	16 CPUs
ADJ	8.13	4.95 (1.64)	3.57 (2.27)	2.98 (2.72)	2.66 (3.05)	2.42 (3.35)	2.22 (3.66)	2.25 (3.61)
ADV	10.15	7.24 (1.40)	6.09 (1.66)	5.52 (1.83)	5.22 (1.94)	4.99 (2.03)	4.79 (2.11)	4.91 (2.06)
LWRAD	15.40	8.49 (1.81)	6.23 (2.47)	5.10 (3.01)	4.68 (3.29)	4.18 (3.68)	3.64 (4.23)	3.31 (4.65)
SWRAD	7.59	4.31 (1.76)	3.27 (2.32)	2.72 (2.79)	2.49 (3.04)	2.32 (3.27)	2.15 (3.53)	2.21 (3.43)
CONVECT	7.48	5.95 (1.25)	5.28 (1.41)	5.00 (1.49)	4.81 (1.55)	4.72 (1.58)	4.64 (1.61)	4.56 (1.64)
DIFF	2.46	1.62 (1.51)	1.29 (1.90)	1.11 (2.21)	1.04 (2.36)	0.98 (2.51)	0.93 (2.64)	1.08 (2.27)
TOTAL	62.83	39.47 (1.59)	30.98 (2.02)	26.98 (2.32)	24.90 (2.52)	23.26 (2.70)	21.83 (2.87)	21.57 (2.91)
MFLOPS	318	506	644	739	801	856	912	923
MEMORY (MWORDS)	5.2	5.2	5.3	5.3	5.3	5.3	5.4	5.5
TOTAL CPU	62.41	76.19	87.04	98.06	111.48	123.37	151.75	280.74

Table 4d: UNIFIED MODEL (New Version with Multitasking Mods)
Dedicated runs of 1 day climate on C916 (Sept 94) Grid = 96 x 73 x 19, Timestep = 30 minutes, Resolution = 300 km
Elapsed Times in Seconds (speed-up)

	1 CPU	2 CPUs	3 CPUs	4 CPUs	5 CPUs	6 CPUs	8 CPUs	16 CPUs
ADJ	8.14	4.61 (1.76)	3.37 (2.41)	2.68 (3.03)	2.40 (3.39)	2.24 (3.63)	1.98 (4.11)	1.88 (4.32)
ADV	9.62	5.00 (1.92)	3.52 (2.73)	2.59 (3.71)	2.13 (4.51)	2.06 (4.66)	1.64 (5.86)	2.34 (4.11)
LWRAD	15.41	7.87 (1.95)	5.27 (2.92)	4.02 (3.83)	3.36 (4.58)	2.78 (5.54)	2.16 (7.13)	1.40 (11.00)
SWRAD	7.58	3.92 (1.93)	2.76 (2.74)	2.22 (3.41)	1.83 (4.14)	1.73 (4.38)	1.39 (5.45)	1.09 (6.95)
CONVECT	7.47	4.88 (1.53)	4.44 (1.68)	3.89 (1.92)	3.74 (1.99)	3.541 (2.12)	3.41 (2.19)	5.09 (1.46)
DIFF	2.45	1.26 (1.94)	0.87 (2.81)	0.67 (3.65)	0.54 (4.53)	0.48 (5.10)	0.41 (5.97)	0.33 (7.42)
TOTAL	62.28	34.45 (1.80)	25.47 (2.44)	20.61 (3.02)	18.04 (3.45)	16.51 (3.77)	14.45 (4.31)	15.44 (4.03)
MFLOPS	323	583	787	970	1107	1208	1378	1288
MEMORY (MWORDS)	5.2	5.4	5.4	5.4	5.4	5.4	5.4	5.6
TOTAL CPU	61.87	67.64	72.75	76.97	82.34	87.82	99.98	162.37

Table 4e: UNIFIED MODEL (Original Version)
Dedicated runs of 6 hour Mesoscale Forecast on C916 (Sept 94) Grid = 92 x 92 x 31, Timestep = 5 minutes, Resolution = 15 km
Elapsed Times in Seconds (speed-up)

	1 CPU	2 CPUs	3 CPUs	4 CPUs	5 CPUs	6 CPUs	8 CPUs	16 CPUs
ADJ	57.42	34.38 (1.67)	23.84 (2.40)	19.94 (2.87)	17.18 (3.34)	15.90 (3.61)	15.02 (3.82)	11.45 (5.01)
ADV	41.90	25.46 (1.64)	19.33 (2.16)	16.45 (2.54)	15.05 (2.78)	14.28 (2.93)	14.05 (2.98)	10.92 (3.83)
LWRAD	33.94	18.36 (1.84)	13.89 (2.44)	11.62 (2.92)	10.27 (3.30)	9.45 (3.59)	8.49 (3.99)	7.78 (4.36)
SWRAD	29.26	16.32 (1.79)	12.44 (2.35)	10.45 (2.80)	9.66 (3.02)	9.08 (3.22)	8.47 (3.45)	8.13 (3.59)
CONVECT	17.33	14.25 (1.21)	12.66 (1.36)	12.19 (1.42)	11.85 (1.36)	11.78 (1.47)	11.88 (1.45)	10.01 (1.73)
DIFF	16.66	12.50 (1.33)	11.14 (1.49)	10.19 (1.63)	10.36 (1.60)	11.26 (1.47)	11.32 (5.97)	8.04 (7.42)
TOTAL	250.84	153.86 (1.63)	117.83 (2.12)	101.64 (2.46)	93.09 (2.69)	89.32 (2.80)	85.81 (2.92)	68.26 (3.67)
MFLOPS	390	636	831	963	1052	1096	1141	1434
MEMORY (MWORDS)	12.8	12.9	12.9	12.9	12.9	12.9	13.0	13.2
TOTAL CPU	248.65	297.14	332.47	373.93	414.69	458.61	568.88	1201.68

Table 4f: UNIFIED MODEL (New Version with multitasking mods)
Dedicated runs of 6 hour Mesoscale Forecast on C916 (Sept 94) Grid = 92 x 92 x 31, Timestep = 5 minutes, Resolution = 15 km
Elapsed Times in Seconds (speed-up)

	1 CPU	2 CPUs	3 CPUs	4 CPUs	5 CPUs	6 CPUs	8 CPUs	16 CPUs
ADJ	58.53	31.32 (1.86)	21.10 (2.77)	16.38 (3.57)	14.09 (4.15)	12.39 (4.72)	9.77 (5.99)	8.04 (7.27)
ADV	39.11	19.88 (1.96)	13.88 (2.81)	10.34 (3.78)	8.86 (4.41)	7.85 (4.98)	6.12 (6.39)	3.91 (10.00)
LWRAD	34.01	17.24 (1.97)	11.62 (2.92)	8.88 (3.82)	7.10 (4.79)	5.99 (5.67)	4.78 (7.11)	2.54 (13.38)
SWRAD	29.32	14.97 (1.95)	10.22 (2.86)	7.96 (3.68)	6.59 (4.44)	5.65 (5.18)	4.73 (6.19)	3.45 (8.49)
CONVECT	17.33	12.99 (1.33)	10.69 (1.62)	9.54 (1.81)	8.77 (1.97)	7.98 (2.19)	7.75 (2.23)	7.92 (2.18)
DIFF	16.69	8.58 (1.94)	5.81 (2.87)	4.33 (3.85)	3.65 (4.57)	3.12 (5.34)	2.33 (7.16)	1.29 (12.93)
TOTAL	249.72	136.99 (1.82)	96.93 (2.57)	77.37 (3.22)	66.49 (3.75)	58.53 (4.26)	49.73 (5.02)	39.12 (6.38)
MFLOPS	396	723	1021	1279	1487	1689	1986	2523
MEMORY (MWORDS)	12.6	12.9	12.7	12.5	12.0	11.9	11.6	11.8
TOTAL CPU	246.79	261.17	272.58	283.70	298.96	313.14	347.61	548.41